Event-Based Time Series Data Preprocessing:

Application to Traffic Flow Time Series

B. Zhu[1], A. Pérez[2], J.P. Valente[2]

CETTICO Research Group. Dept. Lenguajes y Sistemas Informáticos e Ingeniería de Software.
School of Computer Science ETSIInf. Universidad Politécnica de Madrid.
28660 Boadilla del Monte, Madrid, Spain

[1] bo.zhu@alumnos.upm.es
[2] {aurora,jpvalente}@fi.upm.es

Abstract. Traffic flow time series data are usually high dimensional and very complex. Also they are sometimes imprecise and distorted due to data collection sensor malfunction. Additionally, events like congestion caused by traffic accidents add more uncertainty to real-time traffic conditions, making traffic flow forecasting a complicated task. This article presents a new data preprocessing method targeting multidimensional time series with a very high number of dimensions and shows its application to real traffic flow time series from the California Department of Transportation (PEMS web site). The proposed method consists of three main steps. First, based on a language for defining events in multidimensional time series, mTESL, we identify a number of types of events in time series that corresponding to either incorrect data or data with interference. Second, each event type is restored utilizing an original method that combines real observations, local forecasted values and historical data. Third, an exponential smoothing procedure is applied globally to eliminate noise interference and other random errors so as to provide good quality source data for future work.

1 Introduction

Since time series are sequences of numerical values collected regularly at fixed time intervals, they usually contain a huge amount of data. In many domains, such as seismography, electrocardiography, hydrometeorology, etc., as introduced in [2], the relevant information is concentrated in just a few subsequences of the time series and not spread across the whole series. These time series subsequences indicate the occurrence of specific events. In volcano monitoring activities, for example, drastic fluctuations, such as a sudden sharp rise in daily groundwater temperature, may indicate an approaching eruption.

Event definition can also be applied to identify anomalous data. In the domain of traffic flow time series, under normal conditions, traffic flow variation is a stationary random process, which means that the flow data vary within a certain interval only. However, flow data subject to infrequent events like a traffic control or traffic accident are always significantly biased from the normal trend. The occurrence of such events is totally unpredictable, which introduces random deviation and obstructs the illustration of inherent regularity.

1.1 High Dimensional Time Series Event Specification Language (mTESL)

A language for defining events in multidimensional time series is proposed in [1]. Since this language was designed on the basis of concepts of pure mathematics, such as basic set theory, algebra, logic and descriptive statistics, and is a generic means of overcoming domain dependency in the definition of events that appear in time series from different domains. The definition of an event can take into account: sets of interesting points in time series, and other sub-events which need to be established in order to adhere to specific syntax.

In this article we extended this language to very high-dimensional time series so as to reduce the complexity of defining events in such time series. The original language (TESL) treats each dimension in a multidimensional time series as a single time series. But in the particular case of multidimensional time series, exhaustive declarations of all dimensions manually can be extremely time consuming. To solve this problem, we proposed new reserved words such as "ts_set" and "ts_uni" to handle the declaration of dimensions that have similar historical trends using loop processing while preserving the possibility of defining events that involve related dimensions (with posi-

tive/negative correlation, etc.). A new mTESL translator that can translate event definitions to a high-level programming language is under development. It is compatible with Java API and other packages that can be used to include predefined functions to define events more flexibly and easily.

1.2 Exponential Smoothing

Exponential smoothing is a common prediction method. The basic idea is that the predicted value is the weighted sum of previously observed values. Exponential smoothing makes a weighted computation of historical values in chronological order, which means that more weight is attached to recent data than to older data. The interference in the historical trend caused by random factors can be eliminated by applying the exponential smoothing forecasting model. As mentioned in [4], depending on how many times the smoothing process is repeated, it can be classified as simple exponential smoothing, quadratic exponential smoothing, cubic exponential smoothing and higher-order exponential smoothing.

The equation of simple exponential smoothing is as follows:

$$S_t^{(1)} = \alpha * y_t + (1-\alpha) * S_{t-1}^{(1)} \tag{1}$$

where $S_t^{(1)}$ refers to the simple smoothing value of period t, y_t is the real observation value of period t, α refers to the smoothing coefficient, $0 < \alpha < 1$.

The forecasting equation is:

$$x_{t+1} = \alpha * y_t + (1-\alpha) * x_t \tag{2}$$

where x_{t+1} refers to the forecasted value of period t+1.

The choice of α is a crucial factor in the process of exponential smoothing. It should be determined considering the time series variation pattern. A larger value of α approaching 1 can better reflect recent trend changes, whereas a smaller α value close to 0 is less influenced by the real observation and the generated curve is smoother.

Quadratic exponential smoothing repeats the smoothing procedure on the results of simple exponential smoothing. The equation of quadratic exponential smoothing is defined as follows:

$$S_t^{(2)} = \alpha * y_t + (1-\alpha) * S_{t-1}^{(2)} \tag{3}$$

In our research we applied quadratic exponential smoothing. The first iteration restores event areas of time series utilizing exponential smoothing forecasted values. The second iteration generates a smooth curve of target time series. And we proposed a double-sided weighted exponential smoothing step to make full use of historical data.

2 Proposed Method

In general, raw time series data are usually of poor quality and contain mainly two types of suspect data: incorrect data and data with interference. Incorrect data can be further divided into two categories: missing data and distorted data. Missing data are the result of an incomplete data collection process: observation values fail to be recorded at some time stamps because of technical malfunctions or manual operational errors. Distorted data are the result of transient or persistent equipment failure. Data with interference refer to abnormal data with by accidental circumstances interfere. Although they are a true record of observation values, they cannot reflect the inherent either temporal or spatial distribution of the original time series since random circumstances are totally unpredictable, and random error is thus introduced to later research such as classification and clustering etc. In order to eliminate the impact of these poor-quality data, we propose a new event-based time series data preprocessing method, which is outlined in Figure 1. The method consists of the following steps:

Step1: Define the types of events corresponding to erroneous data in multidimensional time series using mTESL.

Step2: Identify missing data and distorted data based on the defined events.

Step3: Restore these erroneous data according to the respective restoration equations using neighboring real observations and historical data. Local forecasted results are generated using an improved quadratic exponential smoothing method.

Step4: Identify abnormal data considering the maximum of delta and mean of n previous observation values before time stamp t, local maximum and event length.

Step5: Restore abnormal data using local forecasted values, real observations and historical data.

Step6: Reapply quadratic exponential smoothing on the whole sequence to preclude random interference.

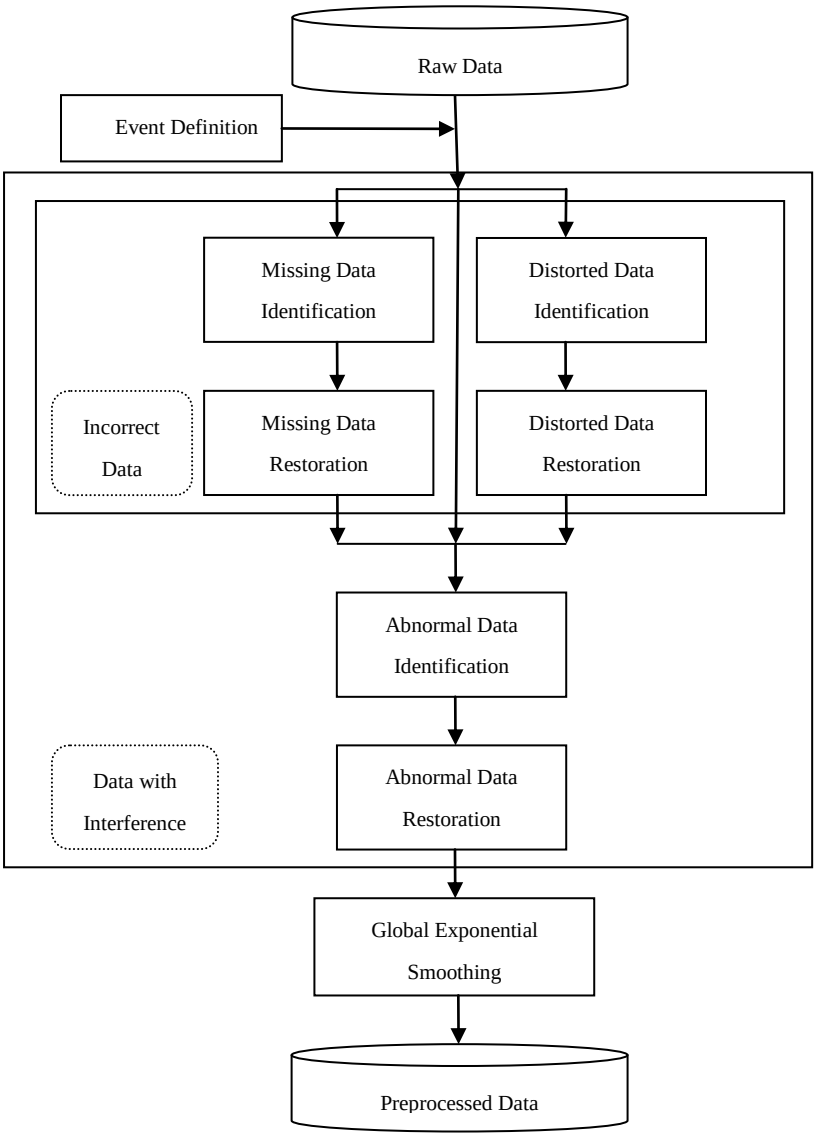

Fig. 1. Proposed Event-based time series data preprocessing method

Here n is the number of previous data taken into consideration. In our case, n is set at 12 by domain experts. This is equivalent to the data from two hours prior to time stamp t. Delta refers to the largest local variation, calculated as local maximum minus local minimum, and mean refers to the local mean of the n previous values. This definition of delta, mean and n applies throughout.

Distorted data and missing data should be processed before abnormal data because the former two are erroneous records and should not be used to restore abnormal data.

3 Application to Traffic Flow Time Series

In our research we applied the proposed preprocessing method to the PEMS-SF data set from the California Department of Transportation web site [6,7]. This data set contains the occupancy rate of different observation stations along San Francisco bay area freeways. Data were recorded in 963 observation stations every 10 minutes over 15 months (440 days). Consequently we can view this data set as 440 high-dimensional time series of 963 dimensions with a length of 144 timestamps for each dimension (6 x 24 = 144).

The PEMS web site has its own mechanism for monitoring sensor performance and diagnosing their real-time health. If a malfunction is diagnosed, the PEMS system refuses to receive more data from that sensor. However, data recorded prior to detecting the malfunction is not deleted.

3.1 Preprocessing using Event definition and identification

Figure 2 illustrates six possible malformed subsequences that may appear in a daily traffic flow time series dimension. These six types of subsequences can be classified into three categories of events considering their consistency with the proposed method: distorted data, missing data and abnormal data. The distorted data can then be further divided into mutations and breakdowns. Given OCC(t), the function value of occupancy at time stamp t, we have defined the following events in order to automate the identification of such incorrect data subsequences:

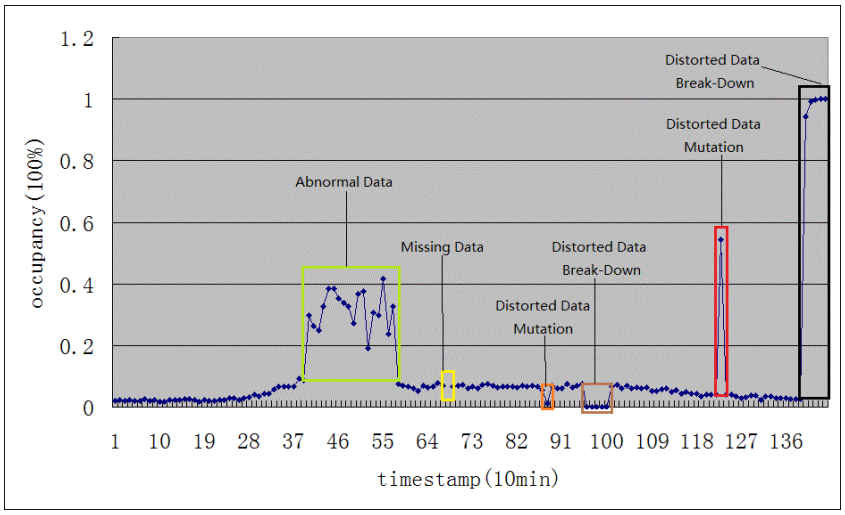

Fig. 2. Illustration of possible malformed subsequences

1. Distorted Data (Mutation).

To identify mutations, we use the local mean E(OCC(t)) and largest delta of n data values before time t. If the occupancy value of time t, OCC(t), satisfies:

OCC(t) < E(OCC(t)) – 2.5 * Delta or OCC(t) > E(OCC(t)) + 2.5 * Delta,

and OCC(t+1) satisfies:

OCC(t+1) < E(OCCt+1) – 2.5 * Delta or OCC(t+1) > E(OCCt+1) + 2.5 * Delta,

then it is considered to be a mutation event. The start of the event is t-1, and the end of the event is t+1.

2. Distorted Data (Breakdown):

A breakdown event is identified when at least five consecutive time stamps have an occupancy value greater than 0.85 (or all zero). The nearest previous normal time stamp is the start of the event, and the nearest subsequent normal time stamp (or the end of the time series if reached) is the end of the event.

3. Missing Data:

The missing data event is easy to identify. The start of the event is t-1, and the end of the event is t+1.

4. Abnormal Data:

The local mean E(OCC(t)) and maximum of delta of n data values before time stamp t are utilized to identify the start of an abnormal data event. If the occupancy value of time t satisfies:

OCC(t) < E(OCC(t)) - 1.5 * Delta or OCC(t) > E(OCC(t)) + 1.5 * Delta,

then the time stamp t is regarded as abnormal, and the previous normal time stamp t-1 with occupancy value OCC(t-1) is taken as the start of the event. And the end of the abnormal data event is the first subsequent time stamp t+h where the occupancy value OCC(t+h) satisfies:

(OCC(t+h)-OCC(t-1))*(OCC(t)-OCC(t-1))<=0.

The maximum occupancy value during the abnormal period should be greater than the predefined threshold (set here at 0.55), and the length of abnormal data should be over 3 time stamps.

The specification of mutation event type using mTESL is as follows:

```
def{
 dimensionset dimensions;
 ts_set set1 TSSET(dimensions);
 ts temp set1(dimensions.foreach);
 basicset tss timestamp(temp);
 stat meantemp mean(temp);
 set candidate
 {  x in tss such that
    temp.value(x) > temp.mean(x,12)+1.5*temp.variation(x,12)
    || temp.value(x)<temp.mean(x,12)-1.5*temp.variation(x,12)};
}
event mutation
 {   peculiar_point in candidate,
      start in tss - candidate,
      end in tss – candidate such that
      start == previous(peculiar_point,tss - candidate)&&
      end == next(peculiar_point,tss - candidate)&&
      (temp.value(end) > temp.mean(peculiar_point,12) + 1.5 *
temp.variation(peculiar_point,12)
```

```
          || temp.value(end) < temp.mean(peculiar_point,12) - 1.5
* temp.variation(peculiar_point,12))};
     }
```

For reasons of space, we will not present the specification of the other events and the details of the extended language mTESL here.

3.2 Data Restoration

We propose a new restoration method and adopt different strategies corresponding to each event type defined above to better restore data. As introduced in [3,5], since missing data and distorted data are not true reflections of traffic flow situations at any time, the observation values are abandoned and not taken into consideration in the restoration step. Instead, we utilize historical data and local forecasted values to restore event subsequence data. Historical data preserve the historical trends of specific time series, which is crucial information for forecasting models. In our case, we adopt the sequence of the same day of the previous week as historical data in order to preserve any possible weekly similarity. And the local forecasted values are generated by an improved quadratic exponential smoothing method.

If OCC(t) is the function of the original occupancy value at time stamp t, FOCC(t) is the forecasted value of time stamp t generated using classic exponential smoothing, ROCC(t) is the function of the occupancy value at time stamp t after restoration, d is the day of the week ($1 \le d \le 7$), HTV(d,t) is the historical trend value of the same day of the previous week, and α is a weighting factor, ($0 \le \alpha \le 1$), the proposed restoration method can be described as:

INPUT: whole sequence, list of inappropriate subsequences (identified events), smoothing coefficient

OUTPUT: whole sequence where inappropriate subsequences are replaced with forecasted subsequences

METHOD:

```
For each event i in the list
   Calculate the event length: Length(i) = end(i) – start(i).
   Calculate the sum of length of all events: SumLength = ∑Length(i)
   IF SumLength / WholeSequenceLength > 0.7
      THEN Replace the whole sequence with historical data HTV(d,t)
      ELSE
```

For each event in the list

For each time stamp

1. From start to end (from left to right)

1.1. Take n previous observation values before start;

1.2. Generate forecasted value FOCC(t) using classic quadratic exponential smoothing model;

1.3. Combine obtained value with historical data HTV(d,t) as follows:

Merger1(t) = α * HTV(d,t) + (1-α) * FOCC(t)

1.4. Use obtained value of Merger1(t)to Generate forecasted value FOCC(t) using classic quadratic exponential smoothing model

For each time stamp

2. From end to start (from right to left)

2.1. Take n previous observation values before end;

2.2. Generate forecasted value FOCC(t) using classic quadratic exponential smoothing model;

2.3. Combine obtained value with historical data HTV(d,t) as follows:

Merger2(t) = α * HTV(d,t) + (1-α) * FOCC(t)

2.4. Use obtained value of Merger2(t)to Generate forecasted value FOCC(t) using classic quadratic exponential smoothing model

3. Forecasted value is calculated as:

ROCC(t) =β* Merger1(t) + (1-β) * Merger2(t)

(β as weighting factor, $(0 \leq \beta \leq 1)$)

For each event type the following particularities have to be taken into account:

- Distorted Data(Mutation): This is a very simple event (only one time stamp) and ROCC is calculated as:

  ROCC(t) = (OCC(t-1) + OCC(t+1))/2

- Distorted Data (Breakdown):

ROCC(t) is calculated as described in the algorithm above.

- Missing Data: In our data set missing data events last only one time stamp:

  ROCC(t) is calculated in the same way as Mutation.

- Abnormal Data:

This type of event is the most complex and longest. The restoration for abnormal data adds one more step to the above algorithm (step 4) making use of real observation values OCC(t):

ROCC(t) =γ* OCC(t) + (1 -γ) * Merger1&2(t)

(γ as weighting factor ($0 \leq \gamma \leq 1$), Merger1&2(t) refers to values generated by step 3 of the algorithm above)

4 Experimental results

The proposed time series data preprocessing method was tested on the PEMS-SF data set. The event identification results are reported in Table 1. Because of the high dimensionality, large volume and abnormal data interference of this data set, many of the events are identified according to event specifications established with the help of domain experts. The occurrence rate for Missing Data and Mutation events is calculated over the total number of timestamps (61015680) as these are single point events. For Breakdown and Abnormal Data events the occurrence rate is calculated over the total number of series (423720) as these are events that can last through several timestamps. Event identification results are consistent with sensor diagnosis and real local traffic flow statistics considering correlations between neighboring dimensions. They also illustrate the need to apply a data preprocessing procedure for raw time series data, since there really are plenty of incorrect data and data with interference, and this will mislead further analysis or data mining investigations.

Table 1. Experiment results of event identification

Events	Occurrences	Occurrence Rate
Missing Data	14912	0.000244
Mutation	99723	0.001634
Break-Down	790	0.001864

Abnormal Data	2013	0.004751

Figure 3 compares the original time series and preprocessing results after different steps. The time series illustrated in Figure 3 is an integration of several real traffic flow time series. Figure 3 shows that the quality of studied time series data has been improved significantly. After data preprocessing, incorrect data that are very biased from the normal trend are restored to a logical range; missing data are filled with estimated values; interference of random errors caused by infrequent events are also reduced; the whole sequence is smoother and noise resistant. Compared with the original time series, preprocessed data preserve seasonal characteristics and trend better.

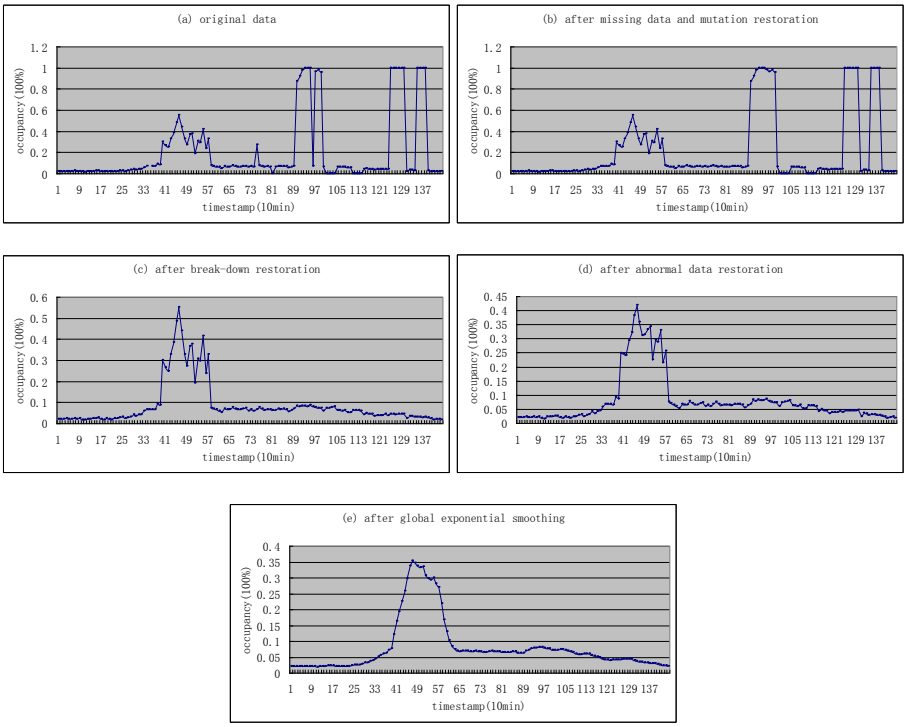

Fig. 3. Original time series and preprocessing results after different steps

Average statistical results for all 963 dimensions in the target data set before and after applying the data preprocessing procedure are shown in Table 2. As we can see, the mean, variance and coefficient of variance changed slightly after preprocessing, which indicates that data preprocessing has preserved the distribution of the majority

of time series well; the kurtosis and skewness values were reduced considerably, which means that the steepness of the spikes and the number of extreme values on the right side of the studied time series both decreased. This suggests that the negative impact caused by extremely high/low values and random errors is removed from the data set.

Table 2. Statistics of PEMS-SF data set before and after preprocessing

	mean	variance	coefficient of variance	kurtosis	skewness
Before Preprocessing	0.0600	0.0028	0.6865	1.4038	0.7597
After Preprocessing	0.0590	0.0022	0.6506	0.1571	0.4855

5 Conclusions and Future Work

High dimensionality and complexity pose a major challenge for time series analysis since the scale of time series data grows exponentially with economic and social development. Random errors introduced by infrequent events and huge amounts of noisy data are also very troublesome for time series analysts. In this article we proposed a data preprocessing model for time series that includes data restoration methods to eliminate noisy data and avoid the interference of random events. We also proposed an extension of the Temporal Event Specification Language (mTESL) to help define and identify events in high dimensional time series. An application to the traffic domain proves that our data preprocessing model is effective. Further investigation of classification based on rough set theory will be conducted on these preprocessed time series data.

References:

1. Juan A. Lara, África López-Illescas, Aurora Pérez, Juan P. Valente. A Language for Defining Events in Multi-Dimensional Time Series: Application to a Medical Domain. 1st International Workshop on Mining of Non-Conventional Data (MINCODA 2009)
2. Juan P. Caraça-Valente, and Ignacio López-Chavarrías, Discovering Similar Patterns in Time Series. KDD 2000, Boston, MA USA, 1-58113-233-6/00/08, 2000.

3. Qi Luo, Transportation Data Analyzing by Using Data Mining Method, Proceedings of the 2008 International Symposiums on Information Processing, Pages 766-767, 2008.
4. Yanyan Zheng, Renzuo Xu, An adaptive exponential smoothing approach for software reliability prediction. Proceedings of 4th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2008).
5. Guiyan Jiang, Longhui jiang, Xiaodong Zhang, Jiangfeng Wang, 动态交通数据故障识别与修复方法 (Malfunction identifying and modifying of dynamic traffic data). Journal of Traffic and Transportation Engineering, 2004, 4(1).
6. California Department of Transportation, http://pems.dot.ca.gov/
7. M. Cuturi, (2011). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.